

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-187742

(43) 公開日 平成10年(1998) 7月21日

(51) Int.Cl. <sup>9</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 17/30			G 0 6 F 15/403	3 3 0 B
12/00	5 1 3		12/00	5 1 3 J
			15/401	3 4 0 C

審査請求 未請求 請求項の数 2 O L (全 11 頁)

(21) 出願番号 特願平8-345899

(22) 出願日 平成8年(1996)12月25日

(71) 出願人 000004226

日本電信電話株式会社

東京都新宿区西新宿三丁目19番2号

(72) 発明者 鈴木 源吾

東京都新宿区西新宿三丁目19番2号 日本

電信電話株式会社内

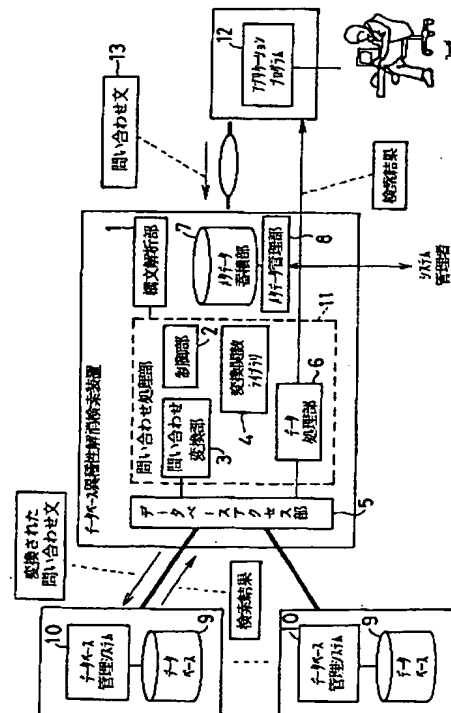
(74) 代理人 弁理士 三好 秀和 (外1名)

(54) 【発明の名称】 データベース異種性解消検索装置

(57) 【要約】

【課題】 各データベースによるデータ項目の表現が同じ意味にも関わらず異なっている、各ユーザ自身の表現形式で各データベースを自由に検索することができるデータベース異種性解消検索装置を提供する。

【解決手段】 ユーザからの問い合わせ文の検索データの記憶されているデータベースおよび該検索データに対応するローカルドメインをローカルドメイン管理テーブルから探索し、該ローカルドメインが含まれるドメイングループからグローバルドメインをドメイングループ管理テーブルから探索し、この探索したローカルドメインとグローバルドメインとが異なる場合、該ローカルドメインをグローバルドメインに変換する変換関数を変換関数管理テーブルから探索し、この変換関数を用いて、各データベースに適合した問い合わせ文に翻訳し、この翻訳された問い合わせ文により各データベースを検索する。



BEST AVAILABLE COPY

## 【特許請求の範囲】

【請求項1】 複数のデータベースが存在する環境において同じ意味を有するが、データベースによってそれぞれ表現が異なるデータ項目をその表現の違いを解消して検索するデータベース異種性解消検索装置であって、各データベースのデータ項目のデータの表現形式であるドメインをローカルドメインとして各データベース毎に管理するローカルドメイン管理テーブルと、同じ意味を有する各ドメインをドメイングループとして管理するとともに、各ドメイングループ内において代表して標準的に使用されるグローバルドメインを選択して管理するドメイングループ管理テーブルと、ドメイン間の変換を行う変換関数を管理する変換関数管理テーブルと、ユーザから入力された問い合わせ文の検索データが記憶されているデータベースおよび該検索データに対応するローカルドメインを前記ローカルドメイン管理テーブルから探索するとともに、該ローカルドメインが含まれるドメイングループからグローバルドメインを前記ドメイングループ管理テーブルから探索するドメイン探索手段と、該ドメイン探索手段で探索したローカルドメインとグローバルドメインとが異なる場合、該ローカルドメインをグローバルドメインに変換する変換関数を前記変換関数管理テーブルから探索する変換関数探索手段と、該変換関数探索手段で探索された変換関数を用いて、前記問い合わせ文を各データベースに適合した問い合わせ文に翻訳し、この翻訳された問い合わせ文により各データベースを検索する検索手段とを有することを特徴とするデータベース異種性解消検索装置。

【請求項2】 前記変換関数探索手段は、前記ローカルドメインをグローバルドメインに直接変換する変換関数を探索できない場合、該ローカルドメインをグローバルドメインに変換しうる複数の変換関数の組合せを探索する手段を有することを特徴とする請求項1記載のデータベース異種性解消検索装置。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】 本発明は、複数のデータベースが存在する環境において同じ意味を有するが、データベースによってそれぞれ表現が異なるデータ項目をその表現の違いを解消して検索するデータベース異種性解消検索装置に関する。

## 【0002】

【従来の技術】 企業は、顧客の管理や、売り上げの管理などさまざまな業務のために、多くのデータベースシステムを導入し、利用している。当初は、顧客や売り上げの情報の記録・変更・利用などのような単純なデータ管理の目的でデータベースを使用していたが、最近は、例えば地域別の商品の売れ筋は何かといった情報を求める

などの経営戦略的な情報を得るためにデータベースを利用することが求められている。

【0003】 そのような目的のためには、1つのデータベースの情報だけでは不十分な場合があり、例えば顧客情報と売り上げ情報を組み合わせるといったように、複数のデータベースの情報を統合して利用する必要がある。しかし、それぞれのデータベースは独立に運用されているために、同じ内容の情報を異なった表現形式で表していることがある。複数のデータベースの情報を統合する場合、その違いをなくすことが必要である。

【0004】 これまで、複数のデータベースを統合して利用する仕組みとして、マルチデータベースが研究・開発されている。マルチデータベースシステムとは、グローバルスキーマと呼ばれるユーザが利用するデータベースの仮想的な構造を予め定義することによって、ユーザは個別の情報の表現形式や構造を意識しないで、分散するデータベースを検索できるというシステムである。

【0005】 しかし、この方式では、ユーザはこのグローバルスキーマの形式でしか、データを見ることができず、1つの見方を強制されることになる。

## 【0006】

【発明が解決しようとする課題】 これまでの方式では、グローバルスキーマを定義するために、すべてのデータベースにおけるデータの表現形式を知っている必要があった。しかし、それには、すべてのデータベースの内容に通じる必要があり、実際は非常に難しい。よって、自分の持つデータベース以外の表現形式をできるだけ知らなくても、データの表現の違いを解消できる仕組みが必要である。

【0007】 また、ユーザは、自分の持っているデータベースと同じ表現形式で、他のデータベースの情報を取りたいという要求がある。これは、自分の使っているアプリケーションを用いて他のデータベースのデータを使いたい、そのアプリケーションが自分のデータベースの表現形式に合わせて作られるからである。この要求を既存のマルチデータベースで実現しようとなると、ユーザのデータベースと同じ形式を持ったグローバルスキーマをユーザ毎に定義する必要があり、非効率であった。よって、グローバルスキーマを定義することなく、断片的にデータの違いを定義することによって、データの表現の違いを解消することが必要である。そして、ユーザが利用したい表現形式を容易に変更できる仕組みが必要である。

【0008】 本発明は、上記に鑑みてなされたもので、その目的とするところは、各データベースによるデータ項目の表現が同じ意味にも関わらず異なっている、各ユーザ自身の表現形式で各データベースを自由に検索することができるデータベース異種性解消検索装置を提供することにある。

## 【0009】

【課題を解決するための手段】上記目的を達成するため、請求項1記載の本発明は、複数のデータベースが存在する環境において同じ意味を有するが、データベースによってそれぞれ表現が異なるデータ項目をその表現の違いを解消して検索するデータベース異種性解消検索装置であって、各データベースのデータ項目のデータの表現形式であるドメインをローカルドメインとして各データベース毎に管理するローカルドメイン管理テーブルと、同じ意味を有する各ドメインをドメイングループとして管理するとともに、各ドメイングループ内において代表して標準的に使用されるグローバルドメインを選択して管理するドメイングループ管理テーブルと、ドメイン間の変換を行う変換関数を管理する変換関数管理テーブルと、ユーザから入力された問い合わせ文の検索データが記憶されているデータベースおよび該検索データに対応するローカルドメインを前記ローカルドメイン管理テーブルから探索するとともに、該ローカルドメインが含まれるドメイングループからグローバルドメインを前記ドメイングループ管理テーブルから探索するドメイン探索手段と、該ドメイン探索手段で探索したローカルドメインとグローバルドメインとが異なる場合、該ローカルドメインをグローバルドメインに変換する変換関数を前記変換関数管理テーブルから探索する変換関数探索手段と、該変換関数探索手段で探索された変換関数を用いて、前記問い合わせ文を各データベースに適合した問い合わせ文に翻訳し、この翻訳された問い合わせ文により各データベースを検索する検索手段とを有することを要旨とする。

【0010】請求項1記載の本発明にあつては、ユーザから入力された問い合わせ文の検索データが記憶されているデータベースおよび該検索データに対応するローカルドメインをローカルドメイン管理テーブルから探索するとともに、該ローカルドメインが含まれるドメイングループからグローバルドメインをドメイングループ管理テーブルから探索し、この探索したローカルドメインとグローバルドメインとが異なる場合、該ローカルドメインをグローバルドメインに変換する変換関数を変換関数管理テーブルから探索し、この変換関数を用いて、ユーザからの問い合わせ文を各データベースに適合した問い合わせ文に翻訳し、この翻訳された問い合わせ文により各データベースを検索するため、各データベースにおける検索データの表現形式がユーザ自身の表現形式と異なっても自由に検索することができる。

【0011】また、請求項2記載の本発明は、請求項1記載の発明において、前記変換関数探索手段が、前記ローカルドメインをグローバルドメインに直接変換する変換関数を探索できない場合、該ローカルドメインをグローバルドメインに変換しうる複数の変換関数の組合せを探索する手段を有することを要旨とする。

【0012】請求項2記載の本発明にあつては、ローカ

ルドメインをグローバルドメインに直接変換する変換関数を探索できない場合、複数の変換関数を組み合わせ、ローカルドメインをグローバルドメインに変換することができる。

【0013】

【発明の実施の形態】以下、図面を用いて本発明の実施の形態について説明する。

【0014】図1は、本発明の一実施形態に係るデータベース異種性解消検索装置の構成を示すブロック図である。本実施形態のデータベース異種性解消検索装置は、複数のデータベースが存在する環境において同じ意味を有するが、データベースによってそれぞれ表現が異なるデータ項目をその表現の違いを解消して検索するものであり、データベースのデータ項目の値の表現形式（これをドメインと称する）を管理し、同じ意味を有するドメインをドメイングループとし、このドメイングループの中から代表して標準的に使用されるグローバルドメインを選択し、検索時にデータベースのローカルドメインとグローバルドメインとが異なる場合、ドメイン間の変換関数を呼び出し、この変換関数を用いてユーザの問い合わせ文を各データベースに適合した問い合わせ文に翻訳し、この翻訳した問い合わせ文によりデータベースを検索するものであり、該データベース異種性解消検索装置は、図1に示すように、ユーザのアプリケーションプログラム12から入力される問い合わせ文13の構文を解析する構文解析部1と、この問い合わせ文に対する処理の全体の流れを制御する問い合わせ処理制御部2、問い合わせの変換を行う問い合わせ変換部3、ドメイン間のデータ変換を行う関数の集まりである変換関数ライブラリ4、および検索されたデータのドメインを変換するデータ処理部6からなる問い合わせ処理部11と、分散データベースにアクセスするためのデータベースアクセス部5と、ドメイン、ドメイングループなどの情報、データベースに含まれるテーブルの情報などを記憶し管理するメタデータ蓄積部7と、該メタデータ蓄積部7に対するメタデータの投入変換などを行うメタデータ管理部8とから構成され、前記データベースアクセス部5にはそれぞれデータベース9および該データベース9を管理するデータベース管理システム10からなる複数のデータベースシステムが接続されている。なお、データベース9はローカルデータベースとも称する。

【0015】このように構成される本実施形態のデータベース異種性解消検索装置は、検索を実行する前にドメインの定義などの準備を行う準備フェーズと実際の検索を行う検索フェーズの2段階の処理を実施する。

【0016】まず、準備フェーズについて説明する。この準備フェーズでは、ドメインの定義を行うが、本実施形態におけるドメインの管理の考え方について図2を参照して説明する。データベースにおけるデータ項目のデータ値の表現法であるドメインとして、例えば図2に示

10

20

30

40

50

すように、血液型を表す場合のドメインとしては、血液型値で表す場合、血液型コードで表す場合、血液型詳細で表す場合などのように種々の表現方法がある。このように種々の表現法の中で同じ意味を有するものをドメイングループとして定義し、この中からただ1つのドメインをユーザに見えるドメインとして設定することで、ユーザに対して単一な見方を提供することができる。なお、図2では、ドメイングループとして、年月日も挙げ、この中には年／月／日や月、日、年などを種々のドメインを示している。また、図2では、ユーザは血液型コードおよび年／月／日を使用しており、これらがユーザドメインとなる。

【0017】ドメインについては、次に示す4つの設定、すなわちローカルドメインの設定、ドメイングループの設定、変換関数の作成、ユーザドメインの設定が必要であり、ユーザはメタデータ管理部8を用いてドメインなどの情報の設定を行う。また、これらの情報はメタデータ蓄積部7に保存される。更に、メタデータ蓄積部7に保存される各情報のテーブルの構成を図3および図4に示す。各ドメインの設定について説明する。

【0018】(1) ローカルドメインの設定：ローカルデータベース9の項目に対し、その表現形式を表す概念である「ドメイン」を設定する。これをローカルドメインと呼ぶ。例えば、あるデータベースのデータ項目「氏名」(型はchar(20))に対し、このドメインをNameであると設定する。このNameというドメインの表す意味は、単にchar(20)という型だけではなく、「氏名」のより細かい表現の形式を表す。例えば、このNameは、

- ・姓が先で、名が後である。
- ・姓と名の間隔は、半角スペースである。

【0019】といった表現形式である。列に対するドメインの定義は、図3(a)に示すように「列」テーブルに保持される。

【0020】基本的には、この作業は、ローカルデータベースの管理者が行う。その時、メタデータ管理部8を用いて、既に定義されているドメインを参照する。もし、自分のデータベースのデータ項目のドメインとして使えるドメインが既に定義されているようなら、それを用い、新たにドメインを作成しない。

【0021】(2) ドメイングループの設定：表現の形式は異なっているが、実際には同じ意味を表しているドメインのグループをドメイングループと呼ぶ。このグループに入るドメインは、すべて同じ意味の異なった表現法である。本装置では、ユーザからは、このドメイングループに含まれるドメインはすべて1つのドメインに統一されて見える。

【0022】ローカルデータベースの管理者は、自分が定義したドメインがどのドメイングループに入るのかを判断し、そのドメイングループに入れる処理をする。適当なドメイングループが見つからない場合は、新しいド

メイングループを作成する。もし、既に定義されているドメインであれば、どれかのドメイングループに含まれているので設定をする必要はない。図3(b)にドメイングループを示す。

【0023】ドメイングループの中には、1つの代表して標準的に使われるドメインを設定しておく(この設定は、基本的に本装置の管理者が行う)。そのドメインをグローバルドメインと呼ぶ。図3(c)にグローバルドメインを示す。

【0024】(3) 変換関数作成：これは、既に変換関数が定義されている場合は、必要はない。グローバルドメインとローカルドメインの間の変換関数を一般のプログラミング言語を用いて作成する。作成した関数は、コンパイルし、ダイナミックにリンクできる形式にする。

【0025】メタデータ管理部8を用いて、この変換関数とドメインの関係を登録する。変換関数とドメインの対応、変換関数の所在は、図4に示すように、「変換関数」テーブルに保持される。

【0026】変換関数は、グローバルドメイン→ローカルドメイン、ローカルドメイン→グローバルドメインというように双方向の関数を作る。ただし、変換によっては、双方向に関数を作ることが不可能な場合もある

(例：名前の変換で、「鈴木 源吾」→「鈴木源吾」という変換＝空白の削除は可能だが、その逆は一般的には不可能である)。その場合は、片方向のみの関数を作成する。双方向の関数を作る理由は、直接変換する関数がない場合に、関数を組み合わせて変換を行うためである。

【0027】(4) ユーザドメインの設定：ドメイングループの中から、ユーザに実際に見えるドメインを(ユーザ毎に)選んでおく。これをユーザドメインと呼ぶ。ユーザが本装置に問い合わせを行う場合、さまざまなドメインにより表現されている項目があっても、すべてこのユーザドメインによる表現に統一される。また、このユーザドメインを変更することによって、ユーザからのデータの見え方を変更することができる。図3(d)にユーザドメインを示す。

【0028】次に、検索フェーズについて説明する。ドメインの情報を用いて以下のように検索を行う。ユーザは、一般にアプリケーションプログラムから、本装置の機能をライブラリなどの形で呼び出し使用する。ユーザからの問い合わせは、関係データベースの検索言語であるSQL言語で作成する。

【0029】(1) 問い合わせDBの決定：構文解析部1により問い合わせを解析し、問い合わせに含まれるテーブル・列を含むデータベースをメタデータ蓄積部7の「列」テーブルを検索することによって知る。

【0030】(2) 問い合わせに含まれる列のドメインの検索：問い合わせに含まれる列のドメインを図3

(a)に示す「列」テーブルを検索することによって知

る。

【0031】 (3) ユーザドメインの検索：問い合わせ  
ているユーザとドメインから、図3 (d) に示す「ド  
メイングループユーザ」テーブルを検索することにより、  
ユーザドメインを検索する。

【0032】 (4) 変換関数の探索：ユーザドメインと  
(2) のドメインの間の変換関数を図4 に示す「変換関  
数」テーブルを用いて調べる。A→Bと直接適用できる  
変換関数が見つからない場合、A→C、C→Bと組み合  
わせて変換できないかを探索する。これは、ドメインを  
ノード、ドメイン間の変換関数の存在をエッジにしたグ  
ラフ上の探索問題であり、グラフ理論で検討されている  
アルゴリズムを用いて容易に実現することができる。

【0033】 (5) 問い合わせの翻訳と実行：データベ  
ースアクセス部5に用いるシステムの差で、2つの実現  
方式がある。データベースアクセス部5は、データベ  
ース統合のためのミドルウェアを用いる。

【0034】 実現方式1はデータベースアクセス部5が  
拡張可能データベース的な機能を許す場合である。

【0035】 ユーザが定義した関数を使って、機能を拡  
張できるようなデータベース管理システムを拡張可能デ  
ータベース管理システム（拡張可能DBMS）と呼ぶ。  
これに相当する機能がデータベースアクセス部に用いる  
ことのできる場合、変換関数は、拡張可能DBMSのユ  
ーザ定義関数として実現することができる。そして、そ  
の関数を問い合わせ言語の中で用いることができる。

【0036】 データベースアクセス部に渡す問い合わせ  
を作るために、ユーザの問い合わせの文中に含まれる列  
を、変換関数を含む形に置き換える。

【0037】

SQL文は、

select (検索する列) from (検索するテーブル) where (条件式)

の形式である。検索する列と条件式の列のドメインがユ  
ーザドメインと異なる場合、以下の置き換えを行う。

【0038】 A) (検索する列) を (ローカルドメイン  
→ユーザドメインへの変換関数) ((検索する列)) に  
置き換える。

B) (条件式に含まれる列) を (ローカルドメイン→ユ  
ーザドメインへの変換関数) ((条件式に含まれる  
列)) に置き換える。

【0039】 実現方式2はデータベースアクセス部5が  
関係データベースの機能しか持たない場合である。関係  
データベースと同じ機能でしか、データベースアクセス  
できないミドルウェアでは、拡張可能データベースのよ  
うに、ユーザ定義関数を問い合わせ言語内で用いること  
ができないので、実現方式1のように問い合わせを書き  
換えることでは、変換を実現できない。以下のようにし  
てデータの変換を実現する必要がある。

【0040】 ・問い合わせの中に含まれる値の変換：問

い合わせに含まれる値をユーザドメインからローカルド  
メインに変換する。

・検索結果の変換：データ処理部6が、検索された結果  
の値をローカルドメインからユーザドメインに変換す  
る。

【0041】 それぞれの方式ともに、複数のデータベ  
ースにアクセスを行うが、結果はそれぞれのデータベース  
への検索結果の和をとることとする。

【0042】 次に、具体的に合成変換を行う場合と行わ  
ない場合の2つの例について説明する。図5に使用する  
サンプルデータベース (DB) を示す。ドメインなどの  
メタデータについては、図3および図4の値はこの例に  
沿っている。例1、例2の両方で同じデータベースを用  
いる。2つの例は、準備フェーズの変換関数の定義まで  
は共通である。ユーザドメインの設定から先が異なっ  
ている。2つのデータベース (DB) は以下の形式であ  
る。

【0043】

DB1:

テーブル「社員」

「社員」の列= (氏名、血液型)

「氏名」の表現法= (鈴木 源吾)

「血液型」の表現法= (0, 1, 2, 3)

DB2:

テーブル「社員」

「社員」の列= (氏名、血液型)

「氏名」の表現法= (鈴木 源吾)

「血液型」の表現法= (A, B, O, AB)

つまり、DB1の「氏名」は名字と名前の間に空白をお  
かず、DB2の「氏名」は名字と名前の間に半角の空白  
をおく。DB1の血液型は、コードで表現され、DB2  
の血液型は値で表現されている。

【0044】 まず、例1として、合成変換を行わない場  
合について説明する。例1の準備フェーズとして、上述  
したようにローカルドメイン、ドメイングループ、変換  
関数の定義およびユーザドメインの決定がそれぞれ以下  
のように行われる。

【0045】 (1) ローカルドメインの定義：DB1、  
DB2の管理者は、それぞれ自分のDBの列に対してそ  
のローカルドメインを設定する。DB1の氏名のドメイ  
ンは姓名間のスペースがないので、NameNoSpace とす  
る。DB1の血液型のドメインはコードで表現されてい  
るので、BloodTypeCode とする。DB2の氏名のドメイ  
ンは姓名間にスペースがあるので、NameSpace とする。  
DB2の血液型のドメインは半角の値で表現されている  
ので、BloodTypeValue とする。

【0046】 (2) ドメイングループの定義：ここで  
は、ドメイングループとして

・Name (含まれるドメインは、NameNoSpace, NameSpac

e、グローバルドメインは、NameNoSpace)、  
・BloodType (含まれるドメインは、BloodTypeCode、  
BloodTypeValue、BloodTypeValueLarge、グローバルド  
メインは、BloodTypeValueLarge)  
の2つを設定する。ここで、BloodTypeValueLarge と  
は、(A、B、O、AB)という全角文字で値の入って  
いるドメインである。

【0047】

(3) 変換関数の定義:

それぞれのドメイングループについて、以下の変換関数を定義する。

ドメイングループ: Name

変換関数: NameSpace2NameNoSpace()

名字と名前の間の空白を除去する。

逆変換は、姓と名の間を一般的には判断できないので作れない。

ドメイングループ: BloodType

変換関数: BloodTypeCode2BloodTypeValueLarge()

図6の対応表を用いて変換を行う。

変換関数: BloodTypeValue2BloodTypeValueLarge()

半角→全角変換

変換関数: BloodTypeValueLarge2BloodTypeCode()

図6の対応表を用いて値の変換を行う。

変換関数: BloodTypeValueLarge2BloodTypeValue()

全角→半角変換

\*

(2) 氏名と血液型のドメインのチェック:

DB1: 氏名→NameNoSpace、血液型→BloodTypeCode

DB2: 氏名→NameSpace、血液型→BloodTypeValue

(3) それぞれのドメインが含まれるグループのユーザドメインの検索:

DB1、DB2: 氏名→NameNoSpace、血液型→BloodTypeValueLarge

以下(4)以降は、上述した実現方式1と実現方式2に  
分けて説明する。

※

(4) 変換関数探索:

(検索する列)について、(ローカルドメイン→ユーザドメインへの変換関数)を探す。

DB1: 氏名→変換なし

DB2: 氏名→NameSpace2NameNoSpace()

(条件式に含まれる列)について(ローカルドメイン→ユーザドメインへの変換関数)を探  
す。

DB1: 血液型→BloodTypeCode2BloodTypeValueLarge()

DB2: 血液型→BloodTypeValue2BloodTypeValueLarge()

いずれも関数を組み合わせないで変換可能である。

(5) 問い合わせの翻訳を行う:

DB1: select 氏名 where BloodTypeCode2BloodTypeValueLarge(血液型) = "A"

DB2: select NameSpace2NameNoSpace(氏名)

where BloodTypeValue2BloodTypeValueLarge(血液型) = "A"

(6) 問い合わせ実行: 各DB毎の検索結果は、次のよ  
うになる。

DB1: 鈴木美紀

\* (4) ユーザドメインの決定: この例では、ユーザドメ  
インはグローバルドメインと等しいとする(図3(d)  
にあるユーザgsuzukiが使う場合である)。つまり、

(Nameのユーザドメイン) = NameNoSpace

(BloodTypeのユーザドメイン) = BloodTypeValueLarge

とする。

【0048】次に、例1の検索フェーズについて説明す  
る。◎ユーザの問い合わせを次のようにSQL文に変換  
する。

問い合わせ: select 氏名 where 血液型 = "A"

(1) 血液型と氏名を含むテーブルの検索: 「列」テー  
ブルを検索して、DB1とDB2を発見する。

【0049】

20

※【0050】実現方式1の場合について説明する。

DB2: 町原宏毅

最終的な検索結果はその和をとって、

50 鈴木美紀

町原宏毅  
となる。

＊【0051】次に、実現方式2の場合について説明する。  
＊

(4) 変換関数探索

(条件式に含まれる値)について、(ユーザドメイン→ローカルドメインへの変換関数)を探す。

DB1: 血液型→BloodTypeValueLarge2BloodTypeCode( )

DB2: 血液型→BloodTypeValueLarge2BloodTypeValue( )

(5) 値の変換を行う。

BloodTypeValueLarge2BloodTypeCode ("A")の結果は0となる。

BloodTypeValueLarge2BloodTypeValue ("A")の結果はAとなる。

(6) 問い合わせの翻訳を行う。

※DB1: select氏名 where血液型="0"

【0052】

※ DB2: select氏名 where血液型="A"

(7) 各DBに問い合わせを実行する。

DB1:

鈴木美紀

DB2:

町原 宏毅

(8) 検索結果を変換する。

DB2: 氏名は、ユーザドメインとローカルドメインが異なっているので、値を変換する。

NameSpace2NameNoSpace ("町原 宏毅")の結果は、"町原宏毅"

(9) 検索結果の和をとる。

最終的な検索結果は、

鈴木美紀

町原宏毅

となる。

【0053】次に、例2として、合成変換を行う場合について説明する。DB1を使用するユーザが、DB1の形式で情報を得たいという状況を考える。このとき、ユーザドメインをDB1の型にすることになる(図3にあるユーザsatohが使う場合である)。

【0054】まず、準備フェーズについて説明するが、

(3) までは、例1と同様であるので、次のように

(4)を行う。

【0055】(4) ユーザドメインの決定: NameのユーザドメインをNameNoSpaceとする。BloodTypeのユーザドメインをBloodTypeCodeとする。次に、例2の検索フェーズについて説明する。ユーザの問い合わせを次のようにSQL文に変更する。

問い合わせ: select 氏名 where 血液型 = " O "

(1) 血液型と氏名を含むテーブルの検索:

DB 1 と DB 2 を発見する。

(2) 氏名と血液型のドメインのチェック:

DB 1: 氏名 → NameNoSpace、血液型 → BloodTypeCode

DB 2: 氏名 → NameSpace、血液型 → BloodTypeValue

(3) それぞれのドメインが含まれるグループのユーザドメインの検索:

氏名 → NameNoSpace、血液型 → BloodTypeValue

実現方式 1 の場合について (4) 以降を説明する。

(4) 変換関数検索:

(検索する列) について、(ローカルドメイン → ユーザドメインへの変換関数) を探す。

DB 1: 氏名 → 変換なし

DB 2: 氏名 → NameSpace2NameNoSpace( )

(条件式に含まれる列) について (ローカルドメイン → ユーザドメインへの変換関数) を探す。

DB 1: 血液型 → 変換なし

DB 2: 血液型 → 1 つの関数で変換できないから図 7 に示すように以下のルートで変換する。

ドメイン: BloodTypeValue → BloodTypeValueLarge → BloodTypeCode

関数: BloodTypeValue2BloodTypeValueLarge → BloodTypeValueLarge2BloodTypeCode

(5) 問い合わせの翻訳を行う:

DB 1: select 氏名 where 血液型 = " O "

DB 2: select NameSpace2NameNoSpace (氏名)

where BloodTypeValueLarge2BloodTypeCode

(BloodTypeValue2BloodTypeValueLarge(血液型)) = " O "

(6) 問い合わせ実行:

検索結果は、例 1) と同じである。

次に、実現方式 2 の場合について説明する。

(4) 変換関数探索:

(条件式に含まれる値) について、(ユーザドメイン → ローカルドメインへの変換関数) を探す。

DB 1: 血液型 → 変換なし

DB 2: 血液型 → 1 つの関数で変換できないから以下のルートで変換する。

ドメイン: BloodTypeCode → BloodTypeValueLarge → BloodTypeValue

関数: BloodTypeValueLarge2BloodTypeCode → BloodTypeValue2BloodTypeValueLarge

(5) 値の変換を行う:

BloodTypeValue2BloodTypeValueLarge(BloodTypeValueLarge2BloodTypeCode (" O ")) の結果は A となる。

(6) 問い合わせの翻訳を行う:

DB 1: select 氏名 where 血液型 = " O "

DB 2: select 氏名 where 血液型 = " A "

(7) 以降は例 1 の実現方式 2 の場合と同じである。

【0056】

【発明の効果】以上説明したように、本発明によれば、ユーザから入力された問い合わせ文の検索データが記憶されているデータベースおよび該検索データに対応するローカルドメインをローカルドメイン管理テーブルから

探索するとともに、該ローカルドメインが含まれるドメイングループからグローバルドメインをドメイングループ管理テーブルから探索し、この探索したローカルドメインとグローバルドメインとが異なる場合、該ローカルドメインをグローバルドメインに変換する変換関数を変



換関数管理テーブルから探索し、この変換関数を用いて、ユーザからの問い合わせ文を各データベースに適合した問い合わせ文に翻訳し、この翻訳された問い合わせ文により各データベースを検索するので、各データベースにおける検索データの表現形式がユーザ自身の表現形式と異なっても、ユーザは自分の欲しい表現形式でデータを自由に検索して容易に得ることができ、データベースのデータをより自由に利用することが可能となる。また、グローバルスキーマの定義を必要とせず、表現形式の違いを解消することができる。

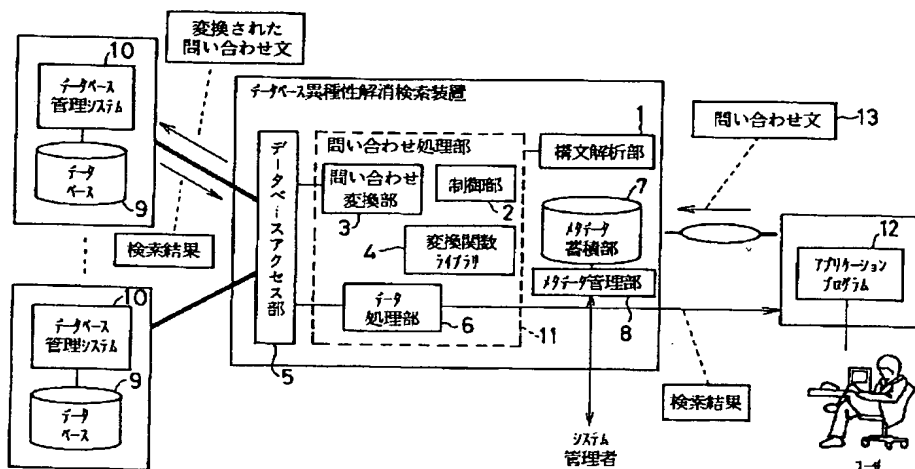
【0057】また、既存のマルチデータベースにおいては、すべてのデータベースにおける表現形式を知らなければ、表現形式の違いを解消できないのに対して、本発明では、ローカルデータベースの管理者が自分の持っているデータベースにおける表現形式であるローカルドメインと標準的な表現形式であるグローバルドメインのみを意識すればよく、他のデータベースでどのように表現されているかを意識する必要がない。

【図面の簡単な説明】

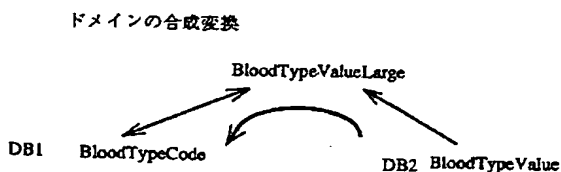
【図1】本発明の一実施形態に係るデータベース異種性解消検索装置の構成を示すブロック図である。

【図2】ドメイン管理の考え方を示す説明図である。

【図1】



【図7】



【図3】図1のデータベース異種性解消検索装置に使用されているメタデータ蓄積部に蓄積される列テーブル、ドメインテーブル、ドメイングループテーブル、ドメイングループユーザテーブルの構成を示す図である。

【図4】図1のデータベース異種性解消検索装置に使用されているメタデータ蓄積部に蓄積される変換関数テーブルの構成を示す図である。

【図5】サンプルのデータベースを示す図である。

【図6】変換関数が使用する血液型対応表の例を示す図である。

【図7】ドメインの合成変換を示す説明図である。

【符号の説明】

- 1 構文解析部
- 2 問い合わせ処理制御部
- 3 問い合わせ変換部
- 4 変換関数ライブラリ
- 5 データベースアクセス部
- 6 データ処理部
- 7 メタデータ蓄積部
- 8 メタデータ管理部
- 9 データベース
- 10 データベース管理システム

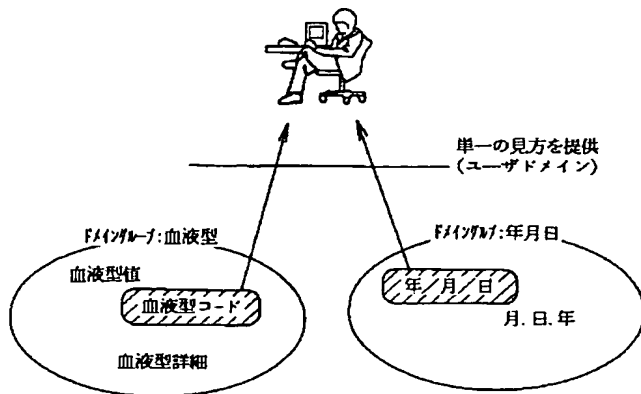
【図6】

BloodTypeCode	BloodTypeValueLarge
0	A
1	B
2	O
3	AB

コード表現

値表現 (金角)

【図2】



【図3】

(a) 「列」テーブル

列名	テーブル名	DB名	ドメイン
氏名	社員	DB1	NameNoSpace
血液型	社員	DB1	BloodTypeCode
氏名	社員	DB2	NameSpace
血液型	社員	DB2	BloodTypeValue

(b) 「ドメイン」テーブル

ドメイン名	ドメイングループ	説明
BloodTypeCode	BloodType	A:0, B:1, O:2, AB:3
BloodTypeValue	BloodType	値による表現 (半角)
BloodTypeValueLarge	BloodType	値による表現 (全角)
NameNoSpace	Name	姓名の順, スペースなし
NameSpace	Name	姓名の順, 半角スペース区切り

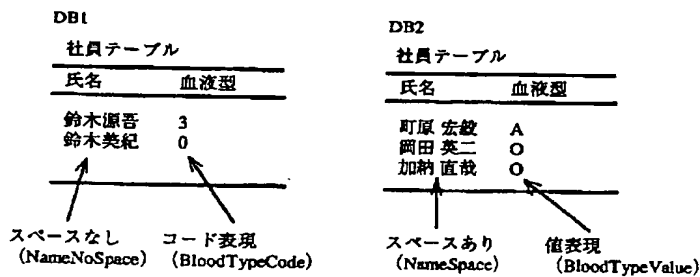
(c) 「ドメイングループ」テーブル

ドメイングループ	グローバルドメイン
BloodType	BloodTypeValueLarge
Name	NameNoSpace

(d) 「ドメイングループユーザ」テーブル

ドメイングループ	ユーザ	ユーザドメイン
BloodType	gsuzuki	BloodTypeValueLarge
BloodType	satoh	BloodTypeCode
Name	gsuzuki	NameNoSpace
Name	satoh	NameNoSpace

【図5】



【図4】

「変換関数」テーブル

fromドメイン	toドメイン	変換関数名	変換関数所在
BloodTypeCode	BloodTypeValueLarge	BloodTypeCode2BloodTypeValueLarge	/usr/local/lib/BloodType.so
BloodTypeValue	BloodTypeValueLarge	BloodTypeValue2BloodTypeValueLarge	/usr/local/lib/BloodType.so
BloodTypeValueLarge	BloodTypeCode	BloodTypeValueLarge2BloodTypeCode	/usr/local/lib/BloodType.so
BloodTypeValueLarge	BloodTypeValue	BloodTypeValueLarge2BloodTypeValue	/usr/local/lib/BloodType.so
NameSpace	NameNoSpace	NameSpace2NameNoSpace	/usr/local/lib/Name.so

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKewed/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.